SID-SLAM: Semi-Direct Information-Driven RGB-D SLAM

Alejandro Fontan[1,2], Riccardo Giubilato[2], Laura Oliva[2], Javier Civera[1] and Rudolph Triebel[2,3]

***Abstract*—This work presents SID-SLAM, a complete SLAM framework for RGB-D cameras. Our main contribution is a semi-direct approach that, for the first time, combines tightly and indistinctly photometric and feature-based image measurements. Additionally, SID-SLAM uses information metrics to reduce the state size with a minimal impact in the accuracy. Our evaluation on several public datasets shows that we achieve state-of-the-art performance regarding accuracy, robustness and computational footprint in CPU real time. In order to facilitate research on semi-direct SLAM, we record the Minimal Texture dataset, composed by RGB-D sequences challenging for current baselines and in which our pipeline excels.**

***Index Terms*—Localization, SLAM**

I. Introduction

Visual odometry and SLAM systems are typically divided in the literature into two categories, *feature-based* and *direct methods*, depending on the type of residuals that are minimized [1]. But, **why should we choose between the two?**

Our contribution in this paper is a strategy to use indistinctly feature-based or photometric residuals, depending only on their information content, and minimizing them jointly to estimate the SLAM state. See Fig. 1 for an illustration of our results. We implemented SID-SLAM, a full RGB-D SLAM pipeline to evaluate our proposal. Our results demonstrate that an information-based tight fusion of photometric and feature-based residuals achieves state-of-the-art performance in accuracy, robustness and computational footprint. Our fusion of residuals is particularly useful for minimal texture cases. In order to illustrate that, we contribute a novel dataset which is conceptually simple, but extremely challenging for current RGB-D SLAM baselines.

The key of our proposal is the complementary nature of feature-based and photometric methods. We will elaborate further on this in the rest of this section. *Features* (e.g., corners, blobs) can be robustly tracked up to a certain degree of illumination and viewpoint changes. However, they appear sparsely in images and hence do not exploit all available information. In contrast, *direct methods* [2] use potentially all available data, since they use the raw pixel intensities. But their high variance to illumination and perspective changes, most often not accounted by the residual models, makes them fragile in practical applications. Rolling shutter effects, sensor asynchronism and calibration errors [3] are also more problematic for direct methods than feature-based ones.

Fig. 1: (a) Features (orange squares) and high-gradient pixels (blue circles) tracked by SID-SLAM and estimated map in a ETH3D sequence [4]. Jointly minimizing photometric and feature-based residuals improves robustness and accuracy, specially in scenes such us our minimal texture dataset (b).

Features require data association, for which correspondences are searched independently first, and robust estimation deals with spurious matches later. Since feature detection and matching runs at real time, most detectors are optimized for speed rather than precision. Direct methods do not need prior data association, since this is implicitly given by the geometry. This allows to track pixels on weak corners and edges, in environments with little or high-frequency textures (e.g., sand [5] or asphalt). However, as a drawback, their convergence is limited to the basin of attraction of image gradients.

An important difference to be highlighted is the geometric dimension of both minimization errors. The error is 2-dimensional in the case of features, whereas the alignment of photometric patches is restricted to the direction of their intensity gradient [6]. Also related to it, the main challenge for a successful fusion is a proper model of the residual covariances, which we address in our work.

II. Related Work

The taxonomy of modern VO/SLAM into *feature-based*, *direct* and *semi-direct* (or *hybrid*) has been extensively addressed in previous works [3], [7], [4]. Here we focus only on

Manuscript received: October, 19, 2022; Revised December, 12, 2022; Accepted February, 6, 2023.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the German Aerospace Center (DLR), the Spanish Government (PID2021-127685NB-I00 and TED2021-131150B-I00) and the Aragon Government (DGA_FSE-T45_20R).

[1]A. Fontan and J. Civera are with School of Engineering, University of Zaragoza, Spain jcivera@unizar.es

[2]A. Fontan, R. Giubilato, L. Oliva and R. Triebel are with German Aerospace Center (DLR), Perception and Cognition Department, Institute of Robotics and Mechatronics, Germany alejandro.fontanvillacampa@dlr.de

[3]R. Triebel is with Department of Informatics, Technical University of Munich, Germany rudolph.triebel@dlr.de

Digital Object Identifier (DOI): see top of this page.

semi-direct methods, that exploit the complementarity of both feature-based and direct methods.

Combining corners and higher-level features. Combining geometric features has been extensively explored. As a few examples, [8], [6] use reprojection errors of corners and edgelets, [9], [10], [11] combine in different manners points and lines, and [12], [13], [14] incorporate points and planes in the map state.

Loose coupling between photometric and feature-based residuals. There are several works in the literature that use features and direct methods in SLAM, but always at different parts of the pipeline and in a loosely coupled manner. [8], [6], [15] used photometric alignment for tracking and pixel triangulation, and feature-based joint optimization of structure and motion. Similarly, [7] combines photometric bundle adjustment of the local structure and motion [2] and geometric bundle adjustment for larger optimization windows [16]. Early direct SLAM algorithms [17], [18], [19] used nearest neighbour search over keyframes for the loop closure. [20], [21] added to a direct VO thread a bag-of-words loop closure [22] and the optimization of a co-visibility graph of keyframe poses. This is similar in [4], but in this case the map optimization is done by an alternating direct Bundle Adjustment. Although all these works benefit from both point types, their loose coupling limits their performance compared to using the same landmarks in tracking, mapping, and relocalization tasks [23], [24]. Up to the authors' knowledge, only the early [25] uses together photometric and image reprojection errors for the case of pairwise camera motion.

Colored ICP. Minimizing 3-dimensional distances together with photometric errors has been used in many RGB-D odometry/SLAM works, e.g., [26], [18], [15]. Differently from us, they use both errors *always* and do not select the most informative one. Their relative weight is tuned experimentally in most cases, which might cause problems in domain changes.

III. SEMI-DIRECT MODEL FORMULATION

Points. We represent 3D map points $\mathbf{p} \in \mathcal{P}\{\boldsymbol{\phi} \cup \mathbf{f}\} \in \mathbb{R}^3$ according to their image representation, that is, $\boldsymbol{\phi} \in \mathbb{R}^3$ if they are represented by image patches, or $\mathbf{f} \in \mathbb{R}^3$ if they are represented by feature descriptors. The image coordinates and inverse depth of $\mathbf{p} \in \mathbb{R}^3$ in reference frame j are denoted as $\mathbf{u}_j \in \boldsymbol{\Omega}$ and $d \in \mathbb{R}$, where $\boldsymbol{\Omega}$ is the image domain. For photometric patches we store a set of intensity values spread in a pattern $\mathcal{N}_\phi$ centered in $\mathbf{u}_j$ [2].

Keyframes. A keyframe j contains RGB-D channels, 6-DoF camera pose as a transformation matrix $\mathbf{T} \in \mathbf{SE}(3)$, two brightness parameters $\{a_j, b_j\}$ and a set of reference points to track. The Lie-algebras pose-increments $\widehat{\mathbf{x}}_{\mathfrak{se}(3)} \in \mathfrak{se}(3)$, with $\widehat{\cdot}_{\mathfrak{se}(3)}$ being the mapping operator from the vector to the matrix representation of the tangent space [27], are expressed as a vector $\mathbf{x} \in \mathbb{R}^6$. During the optimization, we update the transformations at step (k) using left matrix multiplication and the exponential map operator $\exp(\cdot)$, i.e.,

$$\mathbf{T}^{(k+1)} = \exp(\widehat{\mathbf{x}}_{\mathfrak{se}(3)}) \cdot \mathbf{T}^{(k)}. \tag{1}$$

The image points $\mathbf{u}_i$ and $\mathbf{u}_j$ are related by

$$\mathbf{u}_i = \Pi(\mathbf{R}\Pi^{-1}(\mathbf{u}_j, d_j) + \mathbf{t}), \tag{2}$$

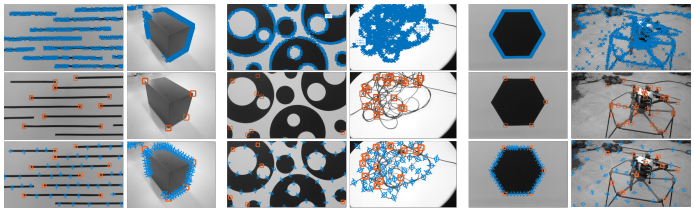


Fig. 2: **Information-based point selection in RGB-D TUM, ETH and our MT dataset.** Top: high-gradient points. Middle: features. Bottom: information-based selection.

where $\Pi(\mathbf{p})$ and $\Pi^{-1}(\mathbf{u}, d)$ are the projection and back-projection functions and $\mathbf{R} \in \mathbf{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the relative rotation and translation between frames.

Residuals. The squared photometric residual $r_i^2|_\phi \in \mathbb{R}$ of a patch $\phi \in \mathcal{P}$ is the sum of the squared intensity differences between all pixels $\mathbf{u}_j$ in the pattern $\mathcal{N}_\phi$ projected in frame i, with the corresponding pixel intensities in its reference keyframe j, combined with a logarithmically parametrized scalar factor e^{-a} and a photometric bias b [2],

$$r_\phi^2|_i = \sum_{\mathbf{u}_j \in \mathcal{N}_\phi} \left(e^{-a_j}(I_j(\mathbf{u}_j) - b_j) - e^{-a_i}(I_i(\mathbf{u}_i) - b_i) \right)^2. \tag{3}$$

The reprojection residual $\mathbf{r}_i|_f \in \mathbb{R}^2$ of a feature $f \in \mathcal{P}$ in frame i is the geometric difference between the landmark projection $\mathbf{u}_i$ and its associated observation, $\hat{\mathbf{u}}_i$

$$\mathbf{r}_f|_i = \hat{\mathbf{u}}_i - \mathbf{u}_i. \tag{4}$$

Residual Covariances. We use the model in [28] to properly model the multi-view covariances $\sigma_\phi^2 \in \mathbb{R}$ and $\boldsymbol{\sigma}_f^2 \in \mathbb{R}^2$ of the residuals in equations (3) and (4). The photometric covariance

$$\sigma_\phi^2 = \sigma_I^2 + G^2 \sigma_\varphi^2(\boldsymbol{\eta}_g, \varepsilon^2, d) \tag{5}$$

is a function of the image noise $\sigma_I^2 \in \mathbb{R}$ and a geometric term σ_φ^2, propagated with the photometric gradient G, which depends on the gradient direction $\boldsymbol{\eta}_g \in \mathbb{R}^2$, the perspective deformation $\varepsilon^2 \in \mathbb{R}$ and the inverse depth $d \in \mathbb{R}$ (see [28] for details). Similarly the covariance of a reprojection residual

$$\boldsymbol{\sigma}_f^2 = \boldsymbol{\sigma}_u^2 + \boldsymbol{\sigma}_\varphi^2(\boldsymbol{\varepsilon}^2, d) \tag{6}$$

depends on the associated noise of the feature descriptor $\boldsymbol{\sigma}_u^2 \in \mathbb{R}^2$ and the propagated geometric noise $\boldsymbol{\sigma}_\varphi^2 \in \mathbb{R}^2$.

A. Informative Point Selection

We extend the approach in [29] to select, in an iterative manner, the most informative points in an image. We analyze the contribution of each point $\mathbf{p}$ to the accuracy of the camera pose $\mathbf{x}$ in terms of entropy reduction [30]:

$$E(\mathbf{x}) = \frac{1}{2}\log_2 \frac{(2\pi e)^k}{|\boldsymbol{\Lambda}_x|}, \quad \Delta_p E(\mathbf{x}) = \frac{1}{2}\log_2(1 + \frac{\Delta_p |\boldsymbol{\Lambda}_x|}{|\boldsymbol{\Lambda}_x|}), \tag{7}$$

where k is the dimension of the information matrix

$$\boldsymbol{\Lambda}_x = \boldsymbol{\Sigma}_x^{-1} = \sum_{\phi \in \mathcal{Q}} \mathbf{j}_\phi^T \sigma_\phi^{-2} \mathbf{j}_\phi + \sum_{f \in \mathcal{Q}} \mathbf{j}_f^T \boldsymbol{\sigma}_f^{-2} \mathbf{j}_f \in \mathbb{R}^{6\times 6} \tag{8}$$

that is the inverse covariance matrix ($\mathbf{\Sigma}_x^{-1} \in \mathbb{R}^{6\times6}$), obtained as the sum of the Jacobian auto-product for the whole set of selected points Q. ($\mathbf{j}_\phi \in \mathbb{R}^{1\times6}$) is the Jacobian of the photometric residual (3) with respect to $\mathbf{x}$. Analogously, ($\mathbf{j}_f \in \mathbb{R}^{2\times6}$) is the Jacobian of the features' residual (4). The variation to the information matrix determinant yielded by the addition of a photometric patch results in [29]

$$\Delta_\phi|\mathbf{\Lambda}_x| = \sigma_\phi^{-2}\mathbf{j}_\phi|\mathbf{\Lambda}_x|\mathbf{\Lambda}_x^{-1}\mathbf{j}_\phi^T, \tag{9}$$

which can be expressed individually per point, depending on $\mathbf{j}_\phi$ and the current inverse covariance matrix. From a first-order Taylor expansion of the determinant of the covariance matrix, we also estimate the contribution to the differential entropy for every feature

$$\Delta_f|\mathbf{\Lambda}_x| \approx |\mathbf{\Lambda}_x|\mathbf{\Lambda}_x^{-1} \cdot (\mathbf{j}_f^T\boldsymbol{\sigma}_f^{-2}\mathbf{j}_f). \tag{10}$$

We select iteratively points that maximize the trade off between their contribution to the camera pose entropy and their spreading in the image

$$s(\boldsymbol{\phi}|\mathbf{f})\Big|_k = \frac{\Delta_p E(\mathbf{x})}{\Delta_p E(\mathbf{x})|_{k=1}} + w \cdot \frac{d_\Omega}{\max d_\Omega|_k}, \tag{11}$$

where d_Ω is the distance for every point with respect to the closest already added point. Since entropy is a scene dependent metric, the first addend normalizes the contribution with the value obtained in the first iteration k. The second normalizes the image distances of the points with the their maximum value on each iteration. We evaluated this selection method on a wide array of scenes (see Figure 2). Figure 3 shows the influence of the relative weight w in the point selection. We set the parameter $w = 0.5$ which balances between entropy maximization and the distribution of points in the image.

B. Information-based tracking

We track every frame reprojecting the points from a local map. We compute the normalized tracking information available per visible point from a reference keyframe with

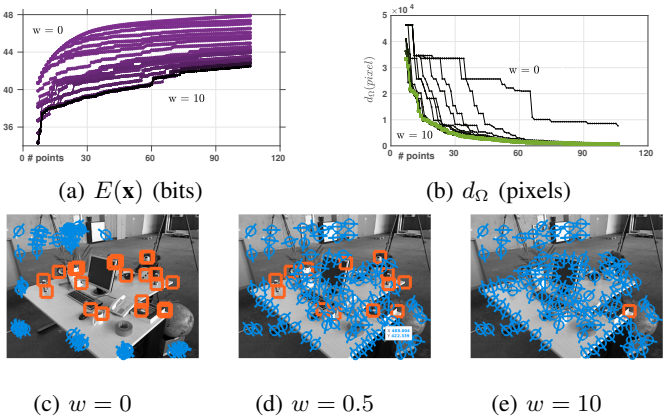
$$\begin{aligned}\overline{E}(\mathbf{x}) = \log_2\left|\left(\frac{\#\phi_r + \#f_r}{\#\phi_w + \#f_w}\right)\mathbf{\Lambda}_x\right| = \\ k\log_2\left(\frac{\#\phi_r + \#f_r}{\#\phi_w + \#f_w}\right) + \log_2(|\mathbf{\Lambda}_x|),\end{aligned} \tag{12}$$

where k is the dimension of ($\mathbf{\Lambda}_x \in \mathbb{R}^{6\times6}$), $\#\phi_r$ and $\#f_r$ are the amount of visible points from the reference keyframe and $\#\phi_w$ and $\#f_w$ the total amount of visible points from the local map. Figure 4 shows how $\overline{E}(\mathbf{x})$ is used as a single threshold for keyframe insertion.

C. Bundle Adjustment with Semi-Direct Formulation

Semi-Direct joint residual. The full combined cost over all frames and points is given by

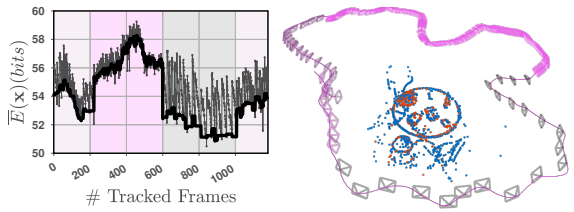
$$\begin{aligned}\sum_{j\in\mathcal{K}}\sum_{\phi\in\mathcal{P}_j}\sum_{i\in obs(\phi)}\left\|\alpha^2(\sigma_i^{-2}r_i^2)_\phi\right\|_\gamma + \\ \sum_{j\in\mathcal{K}}\sum_{f\in\mathcal{P}_j}\sum_{i\in obs(f)}\left\|\beta^2(\boldsymbol{r}_i\boldsymbol{\sigma}_i^{-2}\boldsymbol{r}_i)_f\right\|_\gamma,\end{aligned} \tag{13}$$



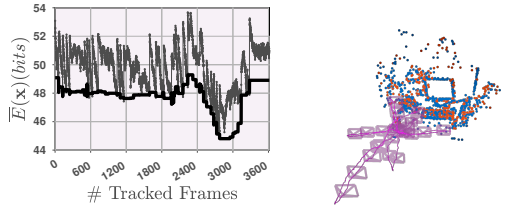
(a) $E(\mathbf{x})$ (bits) (b) d_Ω (pixels)

(c) $w = 0$ (d) $w = 0.5$ (e) $w = 10$

Fig. 3: **Point Selection Strategy.** 3a and 3b show the variation, in consecutive steps of the algorithm, of the entropy of the camera pose $E(\mathbf{x})$ and the minimum image distance d_Ω between iteratively selected points for an interval of values of $w \in [0, 10]$. Note how increasing the influence of the d_Ω addend degrades the entropy contribution and vice-versa. Neglecting to spread image points (3c) concentrates them in the most informative areas. Conversely, small values for w (3e) neglect the informative content of each point. A trade-off between the two balances the selection strategy (3d).



(a) Dynamic Scene



(b) Static Scene

Fig. 4: **Information Threshold for Keyframe Insertion. Top:** Exploratory motion. We manually modified four times the threshold, note how allowing bigger information drops reduces the keyframe creation speed. Darker cameras correspond to bigger information losses. **Bottom:** Non-exploratory motion. Although this sequence is longer the camera is not exploring new areas and our information criterion keeps a low number of keyframes. Note how $\Delta\overline{E}(\mathbf{x})$ reduces when the camera moves away from the map.

where j iterates over all keyframes $\mathcal{K}$, ϕ and f over all points $\mathcal{P}$ in keyframe j, and i over all frames $obs(\phi)$ and $obs(f)$ in which the point ϕ or f are visible. We apply a Cauchy robust cost function to decrease the influence of outliers scaled with a gamma probability value $\gamma_{0.95}$ [18][31].

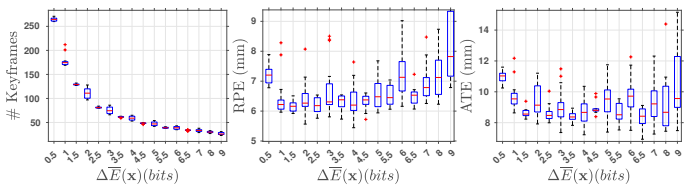


Fig. 5: **Left:** Bigger drops of tracking information $\Delta\overline{E}(\mathbf{x})$ reduce the number of keyframe insertions. **Center:** Decreasing the number of keyframes deteriorates the relative pose error of the tracking. **Right:** The absolute trajectory error has a sweet spot with a 4-bit information drop. Bigger information drops reduce the tracking quality, and lower yield to trajectory drift.

Online covariance correction for residuals. Even if we use sophisticated uncertainty models (see equations (5) and (6)), non-modeled factors (such as motion blur or illumination changes) might unbalance the relative weight between photometric and reprojection residuals and impact the pipeline accuracy. We estimate at run time a correction factor for both residuals iteratively with the covariances α^2, β^2 estimated online from the residual distribution

$$\begin{aligned} \alpha^2 &= \gamma_\phi(\sigma_i^{-2} r_i^2|_{j\in\mathcal{K},\phi\in\mathcal{P},i\in obs(\phi)}), \\ \beta^2 &= \gamma_f(\boldsymbol{r}_i\boldsymbol{\sigma}_i^{-2}\boldsymbol{r}_i|_{j\in\mathcal{K},f\in\mathcal{P},i\in obs(f)}), \end{aligned} \tag{14}$$

where γ_ϕ and γ_f are the functions that map the covariance of gamma distributions from the median of the residuals [18][31].

Motion-only Optimization. As [2], we jointly optimize the camera pose in SE(3) and brightness parameters with a coarse to fine pyramid resolution scheme.

Alternating Full BA. We use alternating optimization between cameras and points instead of jointly optimizing local and global full Bundle Adjustment. This facilitates the real-time performance of photometric Bundle Adjustment by speeding up Gauss-Newton optimization on strongly connected problems [32], [4].

IV. SID-SLAM

A. Windowed optimization

We track each new frame with respect to a reference keyframe and a local window of covisible keyframes around it. As detailed in Section III-B, we insert a new keyframe when the tracking information drops over $\Delta\overline{E}(\mathbf{x})$ in bits. We maintain a graph of covisible keyframes and, for every new keyframe, we trigger a Bundle Adjustment optimization of the cameras and points in a fixed-sized window of the covisibility graph. For additional constraints, we project points of neighbouring keyframes in the optimization window.

Figure 5 shows the relationship between information loss and trajectory errors. The information loss threshold has a clear influence in the keyframes created. Increasing the frequency of keyframe creation frequency improves the tracking quality (observe the RPE trend). However, an excessive number of keyframes reduces their geometric influence and the time available for local Bundle Adjustment. This increases the trajectory drift, and leads to a sweet tuning spot for minimizing the absolute trajectory error (observe the ATE graph). Our experiments allow an information drop of $\Delta\overline{E}(\mathbf{x}) = 4$ (bits) which, as shown in Figure 5, represents a sweet spot in terms of camera trajectory accuracy.

B. Loop Closure, Pose Graph Optimization and Global BA

To correct the pose drift, we implemented a loop closure method using both features and photometric intensities.

Loop detection is performed relying on the full set of AKAZE features extracted for each keyframe. Differently from classical bag-of-words approaches, that require a carefully assembled vocabulary of features, we build upon HBST [33] where a binary tree of feature descriptors is built online and allows for efficient retrieval of similar images from a growing database. Following the insertion of keyframe i, the database is queried for keyframe j such that the number of occurrences of the same visual words is the highest. If the number of matches relative to the total number of features extracted is sufficient, we evaluate the number of co-occurrences with keyframes $j-1$ and $j+1$ looking for temporal consistency. As a first validation step of the candidate match, we match the full set of AKAZE features belonging to keyframes i and j to gather as many correspondences as possible. Then, a classical P3P-RANSAC step returns an initial transformation between the two keyframes.

Loop validation. Semi-Direct alignment is finally conducted as a last barrier against false positives and as a refinement of the estimated transformation, which is then utilized to bootstrap a global Bundle Adjustment step.

Loop closure. Performing alternating *semi-direct* BA instead of a full optimization might be still very costly, and therefore we perform it in three steps. (i) We perform a **pose graph optimization** step which, after a loop detection, aims at pushing old keyframes out of the limited convergence region of the photometric part of the optimization. (ii) We store the geometric reprojection image position for all residuals, either photometric or feature-based, in all previous local windowed optimization. We incorporate to these measures the reprojections obtained in the loop validation step and perform **landmark-like pose BA**. (iii) Finally, we perform **alternating full BA** adjustment to refine the global solution.

V. EXPERIMENTS

Aleatoric effects and real-time constraints make performance comparisons between state-of-the-art SLAM pipelines challenging. In RGB-D SLAM, it is common practice to compare the Absolute Trajectory RMSE with SE(3) alignment (SE(3) ATE RMSE, [34]). Among the good practices, (i) [4] suggests evaluating in **complete benchmarks** instead of subselect (potentially cherry-pick) sequences, [16], [20] compare **median results** (ii) over several runs to account the non-deterministic effects of mutithreading, (iii) [35] shows **memory consumption** (GB) for the captured sequences, and (iv) [6] shows the **processing time** (v) running the experiments in the **same machine**.

In this paper we run our own evaluation of SID-SLAM, the feature-based baselines ORB-SLAM2 [16] and ORB-SLAM3 [24], and the photometric baseline BAD-SLAM [4] in three

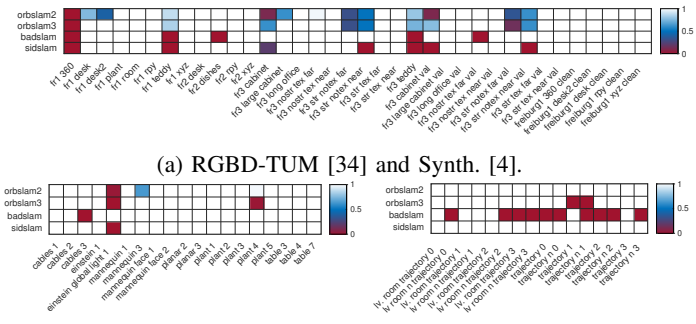
public RGB-D datasets (i). Tables I, II and III gather the median value of the absolute trajectory error over ten runs per sequence (ii). We gather all values found in literature and also the running details. We compare memory footprint and resources consumption using a laptop with an Intel Core i7-10875H, 32 GB of RAM and an NVIDIA GeForce RTX 2070 8GB (iii),(iv),(v). We report the **percentage of trajectory** (vi) that baselines are able to compute. We only compare accuracy for runs that cover **rigorously 100%** (vii) of the sequence, ensuring that accuracy is compared also in challenging parts of the sequences. We collect the values of our evaluation together with those found in the literature and reflect relevant **evaluation conditions** (viii) about each evaluation. We also report the **number of keyframes** (ix) created per sequence which is intimately linked to accuracy and memory footprint. Finally we evaluate in our Minimal Texture dataset.

A. Results in public RGB-D datasets

Accuracy. Figure 6 shows the percentage of trajectory estimated successfully by the three baselines and SID-SLAM. Tables I and II report the ATE for fully completed runs, avoiding misleading comparisons between runs that have been partially estimated. Values in bold represent the smallest tracking error per sequence, values in parentheses correspond to runs with at least 50% of the estimated track, and dashes represent large tracking errors early in the sequence.

RGB-D TUM dataset. Overall, dense methods are more robust than sparse ones in almost textureless sequences, as in *fr3 notex. near* or *fr3 large cabinet*. However, in scenes with low degrees of texture but with visible corners and edges, SID-SLAM makes efficient use of all visual information and outperforms both dense and feature-based methods (as in *fr3 notex. far*). Similarly, in *fr3 tex. far* sequences, where the scene content evolves from a high-frequency textured scene to a gradient-shaped cable, our approach outperforms all the baselines. Even in richly textured sequences as *fr2 desk*, SID-SLAM outperforms the baselines. This is of high merit, as pure feature-based approaches avoid photometric noises and fusion nuisances and they should shine there. In summary, SID-SLAM improves robustness over feature-based methods by completing 24/31 sequences and achieves the best accuracy over all other baselines at 12/31 sequences.

ET3D benchmark and **Synthetic RGB-D TUM**. BAD-SLAM [4] consistently obtains the best accuracy in the ETH3D benchmark [4]. This is due to two factors. Firstly, high-quality sensors that have been calibrated thoroughly downplay the typical filtering effect that features' invariance offer, and that is very convenient with lower quality cameras or poorer calibrations. And secondly, BAD-SLAM's additional minimization of a depth alignment residual. This helps in cases of poor visual information, which is beneficial in this high-quality dataset, but adds a dependency on the depth measurements that might introduce errors in lower-quality data. Our approach achieves similar performance on these sequences which are comparatively shorter (especially *plant* with less than 200 frames) and where the accuracy range is of the order of tenths of a millimeter.



(a) RGBD-TUM [34] and Synth. [4].

(b) ETH3D [4]. (c) ICL-NUIM[36].

Fig. 6: **Percentage of estimated trajectory complete.** SID-SLAM completes **77/89** trajectories, ORB-SLAM2 61/89, ORB-SLAM3 68/89, and BAD-SLAM 61/89.

ICL-NUIM. Our SID-SLAM complete all the sequences and obtains the best accuracy in 6/16 sequences in the living room and office environments. We believe the synthetic nature of the data, with non-informative planar depths in many cases, damaged BAD-SLAM performance.

Keyframe insertion and memory footprint. Figure 7 shows performance differences between the sparse feature-based, sparse semi-direct and dense photometric approaches. The bottom row of the heat map shows the number of inserted keyframes normalized to the number of frames per sequence. As can be observed in the table, our entropy-based criterion inserts the smallest ratio of keyframes, far from the BAD-SLAM ratio of more than 11 keyframes per frame. Note how ORB-SLAM2 and ORB-SLAM3 increase drastically the number of keyframes in some sequences to avoid tracking failure (as in *plant*).

Figure 7 reports the amount of memory allocated per keyframe for each baseline and our SID-SLAM. BAD-SLAM is the most demanding in terms of memory and computation (note that it runs on GPU), as in addition to buffering gray and depth images it handles larger data loads. SID-SLAM uses slightly more memory per keyframe than the feature-based baselines (we need to buffer grey images but not their depth), but this is compensated by the lower keyframe ratio.

B. Results in Minimal Texture Dataset

Motivation. We recorded this new dataset to facilitate research on semi-direct SLAM, particularly on: (i) a better understanding of visual uncertainties of both features and photometric approaches [28], (ii) the efficient use of all the information on the image which maximizes SLAM robustness and reduces its computational footprint [29] [44].

Our dataset consists of 16 conceptually simple but challenging sequences. We group the sequences as *Extreme Geometry* , *Loop*, *Sand*, and *Easy*. The *Easy* set contains control sequences to give an indicative measure of accuracy. *Extreme Geometry* sequences form the core of the dataset, focusing on minimal geometric content and strong perspective changes. The *Loop* set alternates between conceptual geometry content and the laboratory environment. Finally, the *Sand* group is meant to test *semi-direct* SLAM algorithms in textureless

		Freiburg 1							Freiburg 2				Freiburg 3									
	nd.b.	360	desk	desk2	plant	room	rpy	teddy	xyz	desk	dishes	rpy	xyz	cabinet[14]	l. cabinet	l. office	ns.t. far	ns.t. near	s.nt. far	s.nt. near	s.t. far	s.t. near	teddy
KinectFusion [37]	✗[1]	9.1	5.7						2.6							6.4							
RGB-D SLAM [38]	✗[2]	7.9	2.3	4.3	9.1	8.4	2.6	7.6	1.4	9.5[11]		1.9	2.6[11]			3.2[10]		1.7[10]					
MRSMap [17]	✗	6.9	4.3	4.9	2.6	6.9	2.7	3.9	1.3	5.2		2.4	2.0			4.2[10]							
RGB-D SLAM [39]	-		2.6			8.7				5.7													
DVO SLAM [18]	✗	8.3	2.1	4.6	2.8	5.3[12]	**2.0**	**3.4**	1.1	1.7			1.8[10]			3.5		1.8[10]					
Voxel Hashing [40]	✗[3]		2.3							5.7			2.2			2.3		8.7					
Kintinuous [41]	✗[4]		3.7	7.1	4.7	7.5	2.8		1.7	3.4			2.9			3.0		3.1					
Elast. Fusion [42]	✗[5]	10.8	2.0	4.8	**2.2**	6.8	2.5	8.3	1.1	7.1	-	1.5	1.1	-	9.9	1.7	7.4	1.6	3.0	**2.1**	1.3	1.5	**4.9**
ORB-SLAM2 [16]	✔[6]		**1.6**	**2.2**		**4.7**[7]				**0.9**			**0.4**[8]			**1.0**		1.9					
RGBD-TAM [20]	✔[6]	10.1	2.7	4.2	2.5	15.5	2.1	8.1	**1.0**	2.7	**3.6**	**0.2**	0.7	**5.7**	**7.0**	2.7	**2.6**	**1.0**[7]	**1.3**	4.4	1.0	**1.0**	-
Bundle Fusion [35]	✗		**1.6**										1.1			2.2		1.2					
PSM SLAM [43]	✗	**5.5**	**1.6**			5.1										3.1							
BAD-SLAM [4]	✗		1.7[9]										1.1[9]			1.7							
SDTAM [15]	✗		1.8	4.3	3.4	8.4	3.1		1.1				1.5			**1.0**		1.3			**0.9**		
ORB-SLAM2 [16]	✔	-	(1.6)	(2.2)	1.4	**4.3**	2.0	(3.9)	**1.0**	**0.9**	(4.3)	**0.3**	**0.4**	(6.3)	(4.9)	**1.0**	(5.9)	2.4	(0.9)	(2.5)	1.1	1.1	(1.4)
ORB-SLAM3 [24]	✔	-	**1.8**	**2.7**	2.0	7.6	2.2	(6.2)	**1.0**	1.8	8.8	0.5	**0.4**	(2.4)	15.1	1.1	4.6	2.1	(0.9)	(2.3)	**1.0**	**1.0**	(1.2)
BAD-SLAM [4]	✔	-	2.1	**2.7**	**1.1**	8.4	**1.7**	-	**1.0**	6.3	-	0.7	1.1	**1.2**	**3.6**	3.3	8.0	2.7	10.0	**1.0**	2.4	1.4	-
SID-SLAM	✔	-	2.2	3.5	1.4	9.3	3.1	-	**1.0**	**0.9**	**3.6**	0.4	**0.4**	(6.3)	9.6	2.0	**3.2**	**1.9**	**1.8**	-	**1.0**	**1.0**	-

	Freiburg 3 Validation[13]								
	cabinet	l. cabinet	l. office	ns.t. far	ns.t. near	s.nt. far	s.nt. near	s.t. far	s.t. near
ORB-SLAM2 [16]	-	5.8	**0.9**	3.1	1.6	(1.2)	(2.0)	**1.2**	**1.1**
ORB-SLAM3 [24]	-	5.4	1.0	3.5	1.5	(0.9)	(1.6)	**1.2**	**1.1**
BAD-SLAM [4]	**1.1**	**3.1**	2.2	-	3.7	1.8	**6.3**	2.8	1.5
SID-SLAM	-	6.4	1.6	**2.6**	**1.3**	**1.6**	-	1.7	1.3

RGB-D TUM Summary	
	# Best
ORB-SLAM2 [16]	9/31
ORB-SLAM3 [24]	8/31
BAD-SLAM [4]	10/31
SID-SLAM	**12/31**

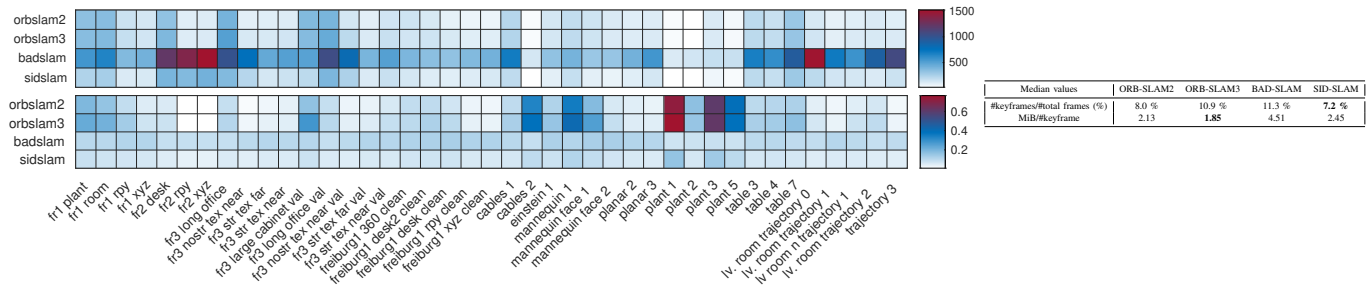
TABLE I: ATE (cm) in the TUM RGB-D benchmark [34] for different baselines got from **their original publications** and from **our own evaluation**. We supply details about how the non-deterministic behavior of the system is account (*nd.b.*). ✗ for papers that do not give information on how they deal with *nd.b.* or that use techniques not suitable for comparison. ✔ for papers that account for *nd.b.*. And ✔ for our evaluation (see Section V for details). **Notes on numbered entries**: (1) Evaluated in [18]. (2) Evaluated in [17]. (3) Evaluated in [35]. (4) Best estimate over ten runs. (5) Using per-sequence best parameters. (6) Values are the median results over 5 runs of each sequence. (7) Typo in [20]. (8) Depth maps were compensated for a 4% bias. (9) This experiments were performed over the RGB and depth distorted images which might explain part of the degraded performance. (10) These values appear in [41] but not in the original publication [18]. (11) These values differ in [41]. (12) Typo in [16]. (13) These sequences were part of the dataset hidden validation but the ground truth is now available. (14) Depths and RGB images are misaligned in the *cabinet* sequence

		ETH3D																			
	nd.b.	cables 1	cables 2	cables 3	cins. 1	cins. glc 1	mann. 1	mann. 3	mann. face 1	man. face 2	planar 2	planar 3	plant 1	plant 2	plant 3	plant 4	plant 5	table 3	table 4	table 7
Elast. Fusion [42]	✔[1]	1.18	1.51	-	2.83	1.11	9.41	-	-	1.0	1.06	3.47	0.80	0.82	1.77	1.17	1.07	-	1.25	-
DVO-SLAM [18]	✔[1]	**0.44**	-	-	0.51	0.86	3.60	2.05	0.52	0.34	**0.24**	9.76	0.84	0.28	0.97	0.29	0.49	0.82	1.82	0.69
Bundle Fusion [35]	✔[1]	2.24	9.51	-	2.89	1.62	-	-	1.34	0.91	0.34	0.35	-	0.41	-	-	-	1.73	-	1.02
ORB-SLAM2 [16]	✔[1]	0.74	0.80	**1.63**	0.40	0.86	**1.19**	1.52	**0.37**	0.11	0.52	5.71	**0.16**	0.28	2.01	0.17	0.41	0.56	0.83	0.99
BAD-SLAM [4]	✔[1]	0.68	**0.51**	6.19	**0.30**	**0.29**	-	**0.55**	0.39	**0.10**	0.30	**0.33**	0.19	**0.14**	**0.18**	**0.14**	**0.17**	**0.24**	**0.22**	**0.28**
ORB-SLAM3 [24]	✗[1,2]	2.12	1.10	2.85	2.19	7.55	4.12	3.28	1.18	7.51	3.36	8.35	0.33	0.65	2.19	0.55	0.35	0.98	1.40	2.43
ORB-SLAM2 [16]	✔	0.77	0.98	2.42	0.48	-	1.29	(2.8)	0.47	0.17	0.35	1.54	0.22	0.24	1.49	(0.24)	0.40	0.52	0.79	0.92
ORB-SLAM3 [24]	✔	0.73	1.08	**2.34**	0.45	-	1.21	1.59	**0.40**	0.23	0.35	1.49	0.27	0.38	1.59	-	0.43	0.53	0.74	1.09
BAD-SLAM [4]	✔	**0.61**	**0.63**	-	**0.32**	**0.69**	**0.44**	**0.44**	0.41	**0.13**	**0.26**	**0.33**	**0.21**	**0.17**	**0.21**	**0.20**	**0.20**	**0.26**	**0.26**	**0.29**
SID-SLAM	✔	0.73	0.80	3.51	0.50	-	1.02	1.76	0.45	0.16	0.30	0.87	0.29	**0.17**	0.43	0.24	0.39	(0.41)[3]	(0.56)[3]	(0.80)[3]

	Synthetic Freiburg (Clean)							avg.[4]		med.[4]	
	360	desk	desk2	rpy	xyz	l. office	ns.t.n.				
ORB-SLAM2 [16]	0.50	0.31	0.31	0.11	0.14	1.31	2.16	0.69	0.47[5]	0.31	0.30 [5]
ORB-SLAM3 [24]	0.72	0.35	0.32	0.13	0.15	1.45	2.45	0.80		0.35	
BAD-SLAM [4]	**0.10**	**0.16**	**0.17**	**0.04**	**0.10**	**0.11**	**0.85**	**0.22**	**0.15**[5]	**0.11**	**0.20**[5,6]
SID-SLAM	0.49	0.34	0.18	0.11	0.17	1.5	0.90	0.53		0.34	

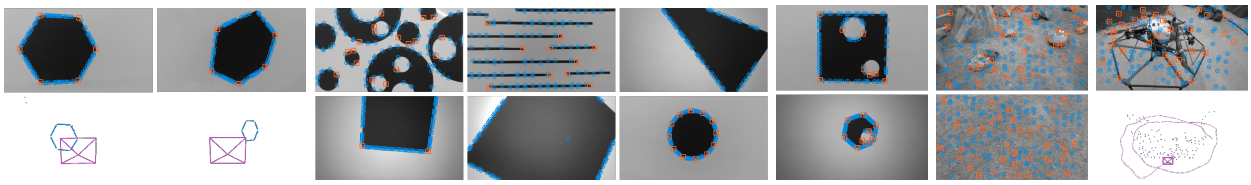
	ICL NUIM															
	lv0	lvn0	lv1	lvn1	lv2	lvn2	lv3	lvn3	traj0	trajn0	traj1	trajn1	traj2	trajn2	traj3	trajn3
ORB-SLAM2 [16]	0.8	0.8	15.2	12.4	**1.7**	2.7	**1.0**	1.1	3.0	4.4	7.3	**5.3**	**1.4**	2.3	8.7	6.0
ORB-SLAM3 [24]	7.6	8.5	19.4	17.1	**1.7**	5.1	2.3	1.7	3.6	10.3	-	-	**1.4**	2.2	7.3	**2.3**
BAD-SLAM [4]	**0.2**	-	**0.2**	**1.1**	7.8	-	-	-	-	-	**0.5**	-	-	-	**0.8**	-
SID-SLAM	0.8	**0.7**	22.3	6.7	2.1	**2.4**	1.2	**1.0**	**2.8**	**3.7**	9.8	6.8	1.6	**1.9**	9.5	4.5

TABLE II: ATE (cm) in the ETH3D benchmark [4], the synthetics RGB-D TUM dataset [4] and ICL-NUIM [36]. We supply details about how the nondeterministic behavior of the system is account *(nd.b.)*. **Notes on numbered entries**: (1) Results from the online leaderboard. (2) This evaluation of ORB-SLAM3 gets distorted results. (3) We get a consistent scale bias of 0.5%. We believe this is due to a misaligment between photometric patches and features on the image that propagates to the range of millimetres. (4) Average and median values of the seven sequences. (5) Values from [4]. (6) Typo in [4].



Median values	ORB-SLAM2	ORB-SLAM3	BAD-SLAM	SID-SLAM
#keyframes/#total frames (%)	8.0 %	10.9 %	11.3 %	**7.2 %**
MiB/#keyframe	2.13	**1.85**	4.51	2.45

Fig. 7: **Memory footprint. Bottom rows**: ratio of #keyframes over #frames per sequence. **Top rows:** Total allocated memory per sequence (MiB). **Table**: median values of the keyframe ratio and the allocated memory per keyframe (Mib).



(a) Strong persp. (b) *Extreme. Geometry* (c) *Loop.* (d) *Sand.* (e) *Easy.*

Fig. 8: Representative frames from the **Minimal Texture dataset**.

	Lines	Circle	Dodecagon	Hexagon	Square	Triangle 1	Triangle 2	Triangle 3	Circ. Dodec.	Circ. Hex.	Circ. Sq.	Rocks 1	Rocks 2	Rocks 3	Ardea	LRU
	Ext.Geometry								**Loop**			**Sand**			**Easy**	
ORB-SLAM2 [16]	1.5	(1.2)	(10.0)	-	-	-	-	-	(13.0)	(10.6)	(6.2)	6.6	-	**3.5**	**1.8**	**5.5**
ORB-SLAM3 [24]	1.5	1.1	(8.9)	1.9[2]	-	-	-	-	(13.7)	-	7.2	6.6	-	3.8	1.9	5.8
BAD-SLAM [4]	-	-	-	-	-	-	-	-	-	-	-	8.0	8.5	3.8	10.0	-
SID-SLAM (ϕ)[1]	-	0.8	2.7	3.0	(2.5)	1.9	3.1	2.4	6.6	7.1	4.7	5.8	7.6	3.9	2.1	6.3
SID-SLAM (f)[1]	2.1	-	-	-	-	-	-	-	-	-	-	-	-	5.1	4.5	7.2
SID-SLAM	**1.1**	**0.9**	**2.3**	**1.4**	**(2.3)**	**1.9**	**3.2**	**2.1**	**6.2**	**10.0**	**4.4**	**5.2**	**7.8**	**3.5**	2.0	5.8

TABLE III: ATE (cm) in **Minimal Texture** for different baselines and SID-SLAM. **Notes on numbered entries**: (1) These values are part of an ablation study and are therefore not suitable to compete with those of state-of-the-art baselines. (2) This is a modified version of ORB-SLAM in which we make it work with a minimum number of features.

scenarios such as with planetary exploration purposes [5]. The dataset was recorded with a Realsense D435i, capturing intensity and depth images of resolution 1920×1080 at 30 Hz. We used a ceiling-mounted Vicon system to record millimeter-level ground truth for the camera pose.

Ablation study. We ablated SID-SLAM in two configurations: using only patches ϕ, and using only features f. The features-only configuration failed in all *Extreme geometry* sequences (*Triangles*, *Square*, *Hexagon* or *Dodecagon*) due to the low number of keypoints which, as can be seen in *Triangle* and *Square* images in Table III, is occasionally reduced to none. Even SID-SLAM fails in *Square* as some configurations are quasi-degenerate. Finally, the patches-only configuration failed in *Lines* because, once again in the image, the photometric gradients were vertically aligned and it was only features placed at the extremes avoiding drift optimization on the horizontal axis. Note that the best accuracy in this sequence is obtained by complete SID-SLAM which grabs the necessary scattered features and refines the solution with photometric vertical gradients.

Evaluation. Table III shows that state-of-the-art baselines, both feature-based and photometric, fail at *Extreme geometry* sequences. This is caused by their inability to extract and process visual information. The reduction of thresholds for feature extraction and matching in ORBSLAM2/3 in sequences with just one *Square* (and thus only four corner-like features) leads to system failure. BAD-SLAM failed in all the geometry sequences. SID-SLAM outperforms all methods significantly both in robustness and accuracy.

VI. CONCLUSIONS

In this work we present SID-SLAM, a complete RGB-D SLAM pipeline that, for the first time, fuses feature-based and direct methods in a tightly-coupled manner. As key contributions of our pipeline, we developed covariance models and information-based procedures for appropriate selection of the most informative points independently of its type and their fusion in a single cost function. We also use information criteria for keyframe selection. A thorough validation on three public datasets solidly demonstrates that SID-SLAM achieves state-of-the-art performance in terms of accuracy, robustness and memory efficiency. We further show the strengths of combining feature-based and direct methods in our novel Minimal Texture dataset, which also illustrates significant limitations in the literature.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[3] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2878–2885, 2018.

[4] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[5] L. Meyer, M. Smíšek, A. Fontan Villacampa, L. Oliva Maza, D. Medina, M. J. Schuster, F. Steidle, M. Vayugundla, M. G. Müller, B. Rebele *et al.*, "The MADMAX data set for visual-inertial rover navigation on Mars," *Journal of Field Robotics*, vol. 38, no. 6, pp. 833–853, 2021.

[6] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[7] S. H. Lee and J. Civera, "Loosely-coupled semi-direct monocular SLAM," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 399–406, 2018.

[8] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*, 2014.

[9] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.

[10] J. P. Company-Corcoles, E. Garcia-Fidalgo, and A. Ortiz, "MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2803–2810, 2022.

[11] L. Zhou, S. Wang, and M. Kaess, "DPLVO: Direct Point-Line Monocular Visual Odometry," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7113–7120, 2021.

[12] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[13] C. Arndt, R. Sabzevari, and J. Civera, "From points to planes-adding planar constraints to monocular SLAM factor graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[14] L. Zhou, D. Koppel, and M. Kaess, "Lidar slam with plane adjustment for indoor environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7073–7080, 2021.

[15] S. Bu, Y. Zhao, G. Wan, K. Li, G. Cheng, and Z. Liu, "Semi-direct tracking and mapping with rgb-d camera for mav," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4445–4469, 2017.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[17] J. Stückler and S. Behnke, "Integrating Depth and Color Cues for Dense Multi-Resolution Scene Mapping Using RGB-D Cameras," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012.

[18] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[19] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European conference on computer vision*, 2014.

[20] A. Concha and J. Civera, "RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2017.

[21] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[22] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[24] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Trans. on Robotics*, 2021.

[25] P. F. Georgel, S. Benhimane, and N. Navab, "A unified approach combining photometric and geometric information for pose estimation." in *British Machine Vision Conference*, 2008.

[26] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *IEEE International Conference on Robotics and Automation*, 2013.

[27] H. Strasdat, "Local accuracy and global consistency for efficient visual slam," Ph.D. dissertation, Department of Computing, Imperial College London, 2012.

[28] A. Fontan, L. Oliva, J. Civera, and R. Triebel, "A model for multi-view residual covariances based on perspective deformation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1960–1967, 2022.

[29] A. Fontan, J. Civera, and R. Triebel, "Information-driven direct RGB-D odometry," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[30] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *2010 IEEE International Conference on Robotics and Automation*, 2010.

[31] R. Gomez-Ojeda, F.-A. Moreno, and J. Gonzalez-Jimenez, "Accurate stereo visual odometry with gamma distributions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[32] L. Platinsky, A. J. Davison, and S. Leutenegger, "Monocular visual odometry: Sparse joint optimisation or dense alternation?" in *2017 IEEE International Conference on Robotics and Automation*, 2017.

[33] D. Schlegel and G. Grisetti, "HBST: A hamming distance embedding binary search tree for feature-based visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3741–3748, 2018.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[35] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[36] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *2014 IEEE international conference on Robotics and automation*, 2014.

[37] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in *10th IEEE international symposium on mixed and augmented reality*, 2011.

[38] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *2012 IEEE International Conference on Robotics and Automation*, 2012.

[39] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE transactions on robotics*, vol. 30, no. 1, pp. 177–187, 2013.

[40] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[41] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[42] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[43] Z. Yan, M. Ye, and L. Ren, "Dense visual SLAM with probabilistic surfel map," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2389–2398, 2017.

[44] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: a survey," *Artificial Intelligence Research and Development*, pp. 363–371, 2008.